

Prim Algoritması Nedir?

Çalışma Kağıdı

Prim algoritması, başlangıçta tek bir köşe ile başlar ve her adımda, henüz ağaca dahil edilmemiş köşelerden birini, mevcut ağaca en düşük ağırlıklı kenar ile bağlayarak büyütür.

Sorular

1. Prim Algoritması, minimum yay ağacını bulurken hangi stratejiyi izler?

- A) Her adımda en kısa kenarı seçerek grafi parçalara ayırır.
- B) Tek bir köşe ile başlar ve her adımda mevcut ağaca en yakın köşeyi ekler.
- C) Tüm kenarları sıralayıp döngü oluşturmayanları seçer.
- D) Grafi rastgele böler ve sonra birleştirir.

2. Prim Algoritması'nın açgözlü olmasının anlamı nedir?

- A) Her adımda global olarak en iyi çözümü seçer.
- B) Her adımda yerel olarak en iyi görünen seçimi yapar.
- C) Tüm olası çözümleri dener.
- D) Rastgele seçimler yapar.

3. Aşağıdakilerden hangisi Prim Algoritması için bir ön koşuldur?

- A) Grafin döngüsüz olması
- B) Grafin yönlü olması
- C) Grafin bağlantılı olması
- D) Grafin tam olması

4. Aşağıdaki graf için Prim Algoritmasını kullanarak bir MST bulunuz.

5. Prim Algoritmasının Adım Adım Uygulanışı

6. Tanımla: Prim Algoritması hangi problemi çözer?

7. Tanımla: Prim Algoritması hangi tür bir algoritmadır?

8. Tanımla: Algoritma nasıl başlar?

Cevap Anahtarı

1. B) Tek bir köşe ile başlar ve her adımda mevcut ağaca en yakın köşeyi ekler. — Prim Algoritması, başlangıçta tek bir köşe ile başlar ve her adımda, henüz ağaca dahil edilmemiş köşelerden birini, mevcut ağaca en düşük ağırlıklı kenar ile bağlayarak büyütür.
2. B) Her adımda yerel olarak en iyi görünen seçimi yapar. — Açgözlü algoritmalar, her adımda o an için en iyi görünen seçimi yaparak ilerlerler. Prim Algoritması da her adımda mevcut ağaca en yakın köşeyi ekleyerek bu stratejiyi izler.
3. C) Grafın bağlantılı olması — Prim Algoritması, minimum yay ağacını bulmak için grafın bağlantılı olmasını gerektirir. Eğer graf bağlantılı değilse, her bir bağlantılı bileşen için ayrı ayrı MST'ler bulunur.
4. 1. Rastgele bir köşe seçin (örneğin A). Ağaca dahil edilen köşeler: {A}. Kenarlar: {}. 2. A'dan çıkıp henüz ağaçta olmayan köşelere giden en kısa kenarı bulun. Bu (A, B) kenarıdır (ağırlık 2). Ağaca dahil edilen köşeler: {A, B}. Kenarlar: {(A, B)}. 3. {A, B} kümelerinden çıkıp henüz ağaçta olmayan köşelere giden en kısa kenarı bulun. Bu (B, C) kenarıdır (ağırlık 3). Ağaca dahil edilen köşeler: {A, B, C}. Kenarlar: {(A, B), (B, C)}. 4. {A, B, C} kümelerinden çıkıp henüz ağaçta olmayan köşelere giden en kısa kenarı bulun. Bu (C, D) kenarıdır (ağırlık 4). Ağaca dahil edilen köşeler: {A, B, C, D}. Kenarlar: {(A, B), (B, C), (C, D)}. 5. Tüm köşeler ağaca dahil edildi. MST'nin toplam ağırlığı: $2 + 3 + 4 = 9$.
5. 1. Boş bir küme (MST kenarları için) ve başlangıç köşesi seçilir. 2. Başlangıç köşesi 'ziyaret edildi' olarak işaretlenir. 3. Tüm komşu kenarların öncelik kuyruğuna eklenmesi. 4. Kuyruktan en düşük ağırlıklı kenarın çekilmesi. 5. Eğer kenarın hedef köşesi henüz ziyaret edilmediyse: a. Kenarın MST'ye eklenmesi. b. Hedef köşenin 'ziyaret edildi' olarak işaretlenmesi. c. Hedef köşenin komşularından çıkan ve henüz ziyaret edilmemiş köşelere giden kenarların kuyruğa eklenmesi. 6. Tüm köşeler ziyaret edilene kadar 4. adımdan devam edilir.
6. Bağlantılı, ağırlıklı bir grafın minimum yay ağacını (MST) bulur.
7. Açgözlü (Greedy) bir algoritmadır.
8. Tek bir köşe ile başlar ve her adımda büyür.

Bounlu

Tüm kartlar, adım adım çözümler ve AI hoca desteği Notek uygulamasında.
Sınav tarihlerini Promy otomatik hatırlatıcıya çevirir.